

IDENTIFYING THE ITEMS MOST RELEVANT TO A CURRENT QUERY BASED ON ITEMS SELECTED IN CONNECTION WITH SIMILAR QUERIES

CROSS-REFERENCE TO RELATED APPLICATION

This application is a continuation-in-part of U.S. Patent Application No. 09/665,822 filed September 20, 2000, which is a continuation-in-part of U.S. Patent Application No. 09/041,081 filed March 10, 1998 now issued as U.S. Patent No. 6,185,558, which is a continuation-in-part of U.S. Patent Application No. 09/033,824 filed March 3, 1998, now abandoned, all of which are hereby incorporated by reference in their entirety.

TECHNICAL FIELD

The present invention is directed to the field of query processing.

BACKGROUND OF THE INVENTION

Many World Wide Web sites permit users to perform searches to identify a small number of interesting items among a much larger domain of items. As an example, several web index sites permit users to search for particular web sites among most of the known web sites. Similarly, many online merchants, such as booksellers, permit users to search for particular products among all of the products that can be purchased from a merchant. In many cases, users perform searches in order to ultimately find a single item within an entire domain of items.

In order to perform a search, a user submits a query containing one or more query terms. The query also explicitly or implicitly identifies a domain of items to search. For example, a user may submit a query to an online bookseller containing terms that the user believes are words in the title of a book. A query server program processes the query to identify within the domain items matching the terms of the query. The items identified by the query server program are collectively known as a query result. In the example, the query result is a list of books whose titles contain some or all of the query terms. The query result is typically displayed to the user as a list of items. This list may be ordered in various ways.

For example, the list may be ordered alphabetically or numerically based on a property of each item, such as the title, author, or release date of each book. As another example, the list may be ordered based on the extent to which each identified item matches the terms of the query.

5 When the domain for a query contains a large number of items, it is common for query results to contain tens or hundreds of items. Where the user is performing the search in order to find a single item, application of conventional approaches to ordering the query result often fail to place the sought item or items near the top of the query result, so that the user must read through many other items in the query result before reaching the
10 sought item. In view of this disadvantage of conventional approaches to ordering query results, a new, more effective technique for automatically ordering query results in accordance with collective and individual user behavior would have significant utility.

Further, it is fairly common for users to specify queries that are not satisfied by any items. This may happen, for example, where a user submits a detailed query that is very
15 narrow, or where a user mistypes or misremembers a term in the query. In such cases, conventional techniques, which present only items that satisfy the query, present no items to the user. When no items are presented to a user in response to issuing a query, the user can become frustrated with the search engine, and may even discontinue its use. Accordingly, a technique for displaying items relating to at least some of the terms in a query even when no
20 items completely match the query would have significant utility.

In order to satisfy this need, some search engines adopt a strategy of effectively automatically revising the query until a non-empty result set is produced. For example, a search engine may progressively delete conjunctive, *i.e.*, ANDed, terms from a multiple term query until the result set produced for that query contains items. This strategy has the
25 disadvantage that important information for choosing the correct items can be lost when query terms are arbitrarily deleted. As a result, the first non-empty result set can be quite large, and may contain a large percentage of items that are irrelevant to the original query as a whole. For this reason, a more effective technique for displaying items relating to at least some of the terms in a query even when no items completely match the query would have
30 significant utility.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 is a high-level block diagram showing the computer system upon which the facility preferably executes.

Figure 2 is a flow diagram showing the steps preferably performed by the facility in order to generate a new rating table.

Figures 3 and 4 are table diagrams showing augmentation of an item rating table in accordance with step 206 (Figure 2).

Figure 5 is a table diagram showing the generation of rating tables for composite periods of time from rating tables for constituent periods of time.

Figure 6 is a table diagram showing a rating table for a composite period.

Figure 7 is a flow diagram showing the steps preferably performed by the facility in order to identify user selections within a web server log.

Figure 8 is a flow diagram showing the steps preferably performed by the facility to order a query result using a rating table by generating a ranking value for each item in the query result.

Figure 9 is a flow diagram showing the steps preferably performed by the facility to select a few items in a query result having the highest ranking values using a rating table.

Figures 10-13 are display diagrams showing examples of considerations used by embodiments of the facility to determine the level of effort expended by the user to select an item from a query result.

DETAILED DESCRIPTION

A software facility ("the facility") for identifying the items most relevant to a current query based on items selected in connection with similar queries is described. The facility preferably generates ranking values for items indicating their level of relevance to the current query, which specifies one or more query terms. The facility generates a ranking value for an item by combining rating scores, produced by a rating function, that each correspond to the level of relevance of the item to queries containing one of the ranking values. The rating function preferably retrieves a rating score for the combination of an item and a term from a rating table generated by the facility. The scores in the rating table

preferably reflect, for a particular item and term, how often users have selected the item when the item has been identified in query results produced for queries containing the term. In some embodiments, the scores also reflect the level of effort users were willing to expend in order to find and select the selected items within query results.

5 In different embodiments, the facility uses the rating scores to either generate a ranking value for each item in a query result, or generate ranking values for a smaller number of items in order to select a few items having the top ranking values. To generate a ranking value for a particular item in a query result, the facility combines the rating scores corresponding to that item and the terms of the query. In embodiments in which the goal is
10 to generate ranking values for each item in the query result, the facility preferably loops through the items in the query results and, for each item, combines all of the rating scores corresponding to that item and any of the terms in the query. On the other hand, in embodiments in which the goal is to select a few items in the query result having the largest ranking values, the facility preferably loops through the terms in the query, and, for each item, identifies the top few rating scores for that term and any item. In some embodiments, the facility uses stemming techniques to incorporate scores for terms having the same roots as the terms in the query. The facility then combines the scores identified for each item to generate ranking values for a relatively small number of items, which may include items not identified in the query result. Indeed, these embodiments of the invention are able to
15 generate ranking values for and display items even in cases in which the query result is empty, *i.e.*, when no items completely satisfy the query.
20

Once the facility has generated ranking values for at least some items, the facility preferably orders the items of the query result in decreasing order of ranking value. The facility may also use the ranking values to subset the items in the query result to a
25 smaller number of items. By ordering and/or subsetting the items in the query result in this way in accordance with collective and individual user behavior rather than in accordance with attributes of the items, the facility substantially increases the likelihood that the user will quickly find within the query result the particular item or items that he or she seeks. For example, while a query result for a query containing the query terms "human" and "dynamic"
30 may contain a book about human dynamics and a book about the effects on human beings of particle dynamics, selections by users from early query results produced for queries containing the term "human" show that these users select the human dynamics book much

more frequently than they select the particle dynamics book. The facility therefore ranks the human dynamics book higher than the particle dynamics book, allowing users, most of whom are more interested in the human dynamics book, to select it more easily. This benefit of the facility is especially useful in conjunction with the large, heterogeneous query results that are typically generated for single-term queries, which are commonly submitted by users.

Various embodiments of the invention base rating scores on different kinds of selection actions performed by the users on items identified in query results. These include whether the user displayed additional information about an item, how much time the user spent viewing the additional information about the item, how many hyperlinks the user followed within the additional information about the item, whether the user added the item to his or her shopping basket, and whether the user ultimately purchased the item. Embodiments of the invention also consider selection actions not relating to query results, such as typing an item's item identifier rather than choosing the item from a query result. Additional embodiments of the invention incorporate into the ranking process information about the user submitting the query by maintaining and applying separate rating scores for users in different demographic groups, such as those of the same sex, age, income, or geographic category. Certain embodiments also incorporate behavioral information about specific users. Further, rating scores may be produced by a rating function that combines different types of information reflecting collective and individual user preferences. Some embodiments of the invention utilize specialized strategies for incorporating into the rating scores information about queries submitted in different time frames.

Figure 1 is a high-level block diagram showing the computer system upon which the facility preferably executes. As shown in Figure 1, the computer system 100 comprises a central processing unit (CPU) 110, input/output devices 120, and a computer memory (memory) 130. Among the input/output devices is a storage device 121, such as a hard disk drive; a computer-readable media drive 122, which can be used to install software products, including the facility, which are provided on a computer-readable medium, such as a CD-ROM; and a network connection 123 for connection the computer system 100 to other computer systems (not shown). The memory 130 preferably contains a query server 131 for generating query results from queries, a query result ranking facility 132 for automatically ranking the items in a query result in accordance with collective user preferences, and item rating tables 133 used by the facility. While the facility is preferably implemented on a

computer system configured as described above, those skilled in the art will recognize that it may also be implemented on computer systems having different configurations.

The facility preferably generates a new rating table periodically, and, when a query result is received, uses the last-generated rating table to rank the items in the query result. Figure 2 is a flow diagram showing the steps preferably performed by the facility in order to generate a new rating table. In step 201, the facility initializes a rating table for holding entries each indicating the rating score for a particular combination of a query term and an item identifier. The rating table preferably has no entries when it is initialized. In step 202, the facility identifies all of the query result item selections made by users during the period of time for which the rating table is being generated. The rating table may be generated for the queries occurring during a period of time such as a day, a week, or month. This group of queries is termed a "rating set" of queries. The facility also identifies the terms of the queries that produced these query results in step 202. Performance of step 202 is discussed in greater detail below in conjunction with Figure 7. In steps 204-208, the facility loops through each item selection from a query result that was made by a user during the time period. In step 204, the facility identifies the terms used in the query that produced the query result in which the item selection took place. In steps 205-207, the facility loops through each term in the query.

In step 206, the facility increases the rating score in the rating table corresponding to the current term and item. Where an entry does not yet exist in the rating table for the term and item, the facility adds a new entry to the rating table for the term and item. Increasing the rating score preferably involves adding an increment value, such as 1, to the existing rating score for the term and item. In some embodiments, the facility may add varying increment values in step 206 depending upon aspects of the current item selection. As one example, some embodiments of the facility make a determination of the amount of effort required by the user to make each selection, and base the increment value added in step 206 on that determination. For example, the selection of a first item that is three times further from the beginning of the query result than is a second item may result in an increment value for the selection of the first item that is three times as large as an increment value for the selection of the second item. Increment values for the selection of items that are reached by traversing additional links may likewise exceed increment values for selections of items that can be displayed without selecting intermediate links. Aspects

relating to the determination of the level of effort required for the user to select an item in a query result are discussed further below in conjunction with Figures 10-13.

In step 207, if additional terms remain to be processed, the facility loops back to step 205 to process the next term in the query, else the facility continues in step 208. In step 208, if additional item selections remain to be processed, then the facility loops back to step 203 to process the next item selection, else these steps conclude.

Figures 3 and 4 are table diagrams showing augmentation of an item rating table in accordance with step 206 (Figure 2). Figure 3 shows the state of the item rating table before its augmentation. It can be seen that the table 300 contains a number of entries, including entries 301-306. Each entry contains the rating score for a particular combination of a query term and an item identifier. For example, entry 302 identifies the score "22" for the term "dynamics" the item identifier "1883823064". It can be seen by examining entries 301-303 that, in query results produced from queries including the term "dynamics", the item having item identifier "1883823064" has been selected by users more frequently than the item having item identifier "9676530409", and much more frequently than the item having item identifier "0801062272". In additional embodiments, the facility uses various other data structures to store the rating scores, such as sparse arrays.

In augmenting the item rating table 300, the facility identifies the selection of the item having item identifier "1883823064" from a query result produced by a query specifying the query terms "human" and "dynamics". Figure 4 shows the state of the item rating table after the item rating table is augmented by the facility to reflect this selection. It can be seen by comparing entry 405 in item rating table 400 to entry 305 in item rating table 300 that the facility has incremented the score for this entry from "45" to "46". Similarly, the facility has incremented the rating score for this item identifier the term "dynamics" from "22" to "23". Although the increment values reflected in the differences between Figures 3 and 4 are both 1, as noted above, different increment values may be used for different item selections. The facility augments the rating table in a similar manner for the other selections from query results that it identifies during the time period.

Rather than generating a new rating table from scratch using the steps shown in Figure 2 each time new selection information becomes available, the facility preferably generates and maintains separate rating tables for different constituent time periods, of a relatively short length, such as one day. Each time a rating table is generated for a new

constituent time period, the facility preferably combines this new rating table with existing rating tables for earlier constituent time periods to form a rating table for a longer composite period of time. Figure 5 is a table diagram showing the generation of rating tables for composite periods of time from rating tables for constituent periods of time. It can be seen in Figure 5 that rating tables 501-506 each correspond to a single day between 8-Feb-98 and 13-Feb-98. Each time a new constituent period is completed, the facility generates a new rating table reflecting the user selections made during that constituent period. For example, at the end of 12-Feb-98, the facility generates rating table 505, which reflects all of the user selections occurring during 12-Feb-98. After the facility generates a new rating table for a completed constituent period, the facility also generates a new rating table for a composite period ending with that constituent period. For example, after generating the rating table 505 for the constituent period 12-Feb-98, the facility generates rating table 515 for the composite period 8-Feb-98 to 12-Feb-98. The facility preferably generates such a rating table for a composite period by combining the entries of the rating tables for the constituent periods making up the composite period, and combining the scores of corresponding entries, for example, by summing them. In one preferred embodiment, the scores and rating tables for more recent constituent periods are weighted more heavily than those in rating tables for less recent constituent periods. When ranking query results, the rating table for the most recent composite period is preferably used. That is, until rating table 516 can be generated, the facility preferably uses rating table 515 to rank query results. After rating table 516 is generated, the facility preferably uses rating table 516 to rank query results. The lengths of both constituent periods and composite periods are preferably configurable.

Figure 6 is a table diagram showing a rating table for a composite period. By comparing the item rating table 600 shown in Figure 6 to item rating table 400 shown in Figure 4, it can be seen that the contents of rating table 600 constitute the combination of the contents of rating table 400 with several other rating tables for constituent periods. For example, the score for entry 602 is "116", or about five times the score for corresponding entry 402. Further, although rating table 400 does not contain an entry for the term "dynamics" and the item identifier "1887650024", entry 607 has been added to table 600 for this combination of term and item identifier, as a corresponding entry occurs in a rating table for one of the other constituent periods within the composite period.

The process used by the facility to identify user selections is dependent upon both the kind of selection action used by the facility and the manner in which the data relating to such selection actions is stored. One preferred embodiment uses as its selection action requests to display more information about items identified in query results. In this
5 embodiment, the facility extracts this information from logs generated by a web server that generates query results for a user using a web client, and allows the user to select an item with the web client in order display additional information about it. A web server generally maintains a log detailing of all the HTTP requests that it has received from web clients and responded to. Such a log is generally made up of entries, each containing information about
10 a different HTTP request. Such logs are generally organized chronologically. Log Entry 1 below is a sample log entry showing an HTTP request submitted by a web client on behalf of the user that submits a query.

1. Friday, 13-Feb-98 16:59:27
2. User Identifier=82707238671
3. HTTP_REFERER=http://www.amazon.com/book_query_page
4. PATH_INFO=/book_query
5. author="Seagal"
6. title="Human Dynamics"

Log Entry 1

It can be seen by the occurrence of the keyword "book_query" in the "PATH_INFO" line 4 of Log Entry 1 that this log entry corresponds to a user's submission of a query. It further
25 can be seen in term lines 5 and 6 that the query includes the terms "Seagal", "Human", and "Dynamics". In line 2, the entry further contains a user identifier corresponding to the identity of the user and, in some embodiments, also to this particular interaction with the web server.

In response to receiving the HTTP request documented in Log Entry 1, the
30 query server generates a query result for the query and returns it to the web client submitting the query. Later the user selects an item identified in the query result, and the web client submits another HTTP request to display detailed information about the selected item. Log Entry 2, which occurs at a point after Log Entry 1 in the log, describes this second HTTP request.

1. Friday, 13-Feb-98 17:02:39
2. User Identifier=82707238671
3. HTTP_REFERER=http://www.amazon.com/book_query
4. PATH_INFO=/ISBN=1883823064

5

Log Entry 2

By comparing the user identifier in line 2 of Log Entry 2 to the user identifier in line 2 of Log Entry 1, it can be seen that these log entries correspond to the same user and time frame.

10 In the "PATH_INFO" line 4 of Log Entry 2, it can be seen that the user has selected an item having item identifier ("ISBN") "1883823064". It can further be seen from the occurrence of the keyword "book_query" on the "HTTP_REFERER" line 3 that the selection of this item was from a query result.

15 Where information about user selections is stored in web server logs such as those discussed above, the facility preferably identifies user selections by traversing these logs. Such traversal can occur either in a batch processing mode after a log for a specific period of time has been completely generated, or in a real-time processing mode so that log entries are processed as soon as they are generated.

20 Figure 7 is a flow diagram showing the steps preferably performed by the facility in order to identify user selections within a web server log. In step 701, the facility positions a first pointer at the top, or beginning, of the log. The facility then repeats steps 702-708 until the first pointer reaches the end of the log. In step 703, the facility traverses forward with the first pointer to the next item selection event. In terms of the log entry shown above, step 703 involves traversing forward through log entries until one is found that
25 contains in its "HTTP_REFERER" line a keyword denoting a search entry, such as "book_query". In step 704, the facility extracts from this item selection event the identity of the item that was selected and session identifier that identifies the user that selected the item. In terms of the log entries above, this involves reading the ten-digit number following the string "ISBN=" in the "PATH_INFO" line of the log entry, and reading the user identifier
30 from the "User Identifier" line of the log entry. Thus, in Log Entry 2, the facility extracts item identifier "1883823064" and session identifier "82707238671". In step 705, the facility synchronizes the position of the second pointer with the position of the first pointer. That is, the facility makes the second pointer point to the same log entry as the first pointer. In step 706, the facility traverses backwards with the second pointer to a query event having a

matching user identifier. In terms of the log entries above, the facility traverses backward to the log entry having the keyword "book_query" in its "PATH_INFO" line, and having a matching user identifier on its "User Identifier" line. In step 707, the facility extracts from the query event to which the second pointer points the terms of the query. In terms of the query log entries above, the facility extracts the quoted words from the query log entry to which the second pointer points, in the lines after the "PATH_INFO" line. Thus, in Log Entry 1, the facility extracts the terms "Seagal", "Human", and "Dynamics". In step 708, if the first pointer has not yet reached the end of the log, then the facility loops back to step 702 to continue processing the log, else these steps conclude.

When other selection actions are used by the facility, extracting information about the selection from the web server log can be somewhat more involved. For example, where the facility uses purchase of the item as the selection action, instead of identifying a log entry describing a request by the user for more information about an item, like Log Entry 1, the facility instead identifies a log entry describing a request to purchase items in a "shopping basket." The facility then traverses backwards in the log, using the entries describing requests to add items to and remove items from the shopping basket to determine which items were in the shopping basket at the time of the request to purchase. The facility then continues traversing backward in the log to identify the log entry describing the query, like Log Entry 2, and to extract the search terms.

Rather than relying solely on a web server log where item purchase is the selection action that is used by the facility, the facility alternatively uses a database separate from the web server log to determine which items are purchased in each purchase transaction. This information from the database is then matched up with the log entry containing the query terms for the query from which item is selected for purchase. This hybrid approach, using the web server logs and a separate database, may be used for any of the different kinds of selection actions. Additionally, where a database separate from the web server log contains all the information necessary to augment the rating table, the facility may use the database exclusively, and avoid traversing the web server log.

The facility uses rating tables that it has generated to generate ranking values for items in new query results. Figure 8 is a flow diagram showing the steps preferably performed by the facility to order a query result using a rating table by generating a ranking value for each item in the query result. In steps 801-807, the facility loops through each item

identified in the query result. In step 802, the facility initializes a ranking value for the current item. In steps 803-805, the facility loops through each term occurring in the query. In step 804, the facility determines the rating score contained by the most recently-generated rating table for the current term and item. In step 805, if any terms of the query remain to be processed, then the facility loops up to step 803, else the facility continues in step 806. In step 806, the facility combines the scores for the current item to generate a ranking value for the item. As an example, with reference to Figure 6, in processing datum having item identifier "1883823064", the facility combines the score "116" extracted from entry 602 for this item and the term "dynamics", and the score "211" extracted from entry 605 for this item and the term "human". Step 806 preferably involves summing these scores. These scores may be combined in other ways, however. In particular, scores may be adjusted to more directly reflect the number of query terms that are matched by the item, so that items that match more query terms than others are favored in the ranking. In step 807, if any items remain to be processed, the facility loops back to step 801 to process the next item, else the facility continues in step 808. In step 808, the facility displays the items identified in the query result in accordance with the ranking values generated for the items in step 806. Step 808 preferably involves sorting the items in the query result in decreasing order of their ranking values, and/or subsetting the items in the query result to include only those items above a threshold ranking value, or only a predetermined number of items having the highest ranking values. After step 808, these steps conclude.

Figure 9 is a flow diagram showing the steps preferably performed by the facility to select a few items in a query result having the highest ranking values using a rating table. In steps 901-903, the facility loops through each term in the query. In step 902, the facility identifies among the table entries for the current term and those entries having the three highest rating scores. For example, with reference to Figure 6, if the only entries in item rating table 600 for the term "dynamics" are entries 601, 602, 603, and 607, the facility would identify entries 601, 602, and 603, which are the entries for the term "dynamics" having the three highest rating scores. In additional preferred embodiments, a small number of table entries other than three is used. In step 903, if additional terms remain in the query to be processed, then the facility loops back to step 901 to process the next term in the query, else the facility continues in step 904. In steps 904-906, the facility loops through each unique item among the identified entries. In step 905, the facility combines all of the scores

for the item among the identified entries. In step 906, if additional unique items remain among the identified entries to be processed, then the facility loops back to step 904 to process the next unique item, else the facility continues in step 907. As an example, if, in item rating table 600, the facility selected entries 601, 602, and 603 for the term "dynamics", and selected entries 604, 605, and 606 for the term "human", then the facility would combine the scores "116" and "211" for the item having item identifier "1883823064", and would use the following single scores for the remaining item identifiers: "77" for the item having item identifier "0814403484", "45" for the item having item identifier "9676530409", "12" for the item having item identifier "6303702473", and "4" for the item having item identifier "0801062272". In step 907, the facility selects for prominent display items having the top three combined scores. In additional embodiments, the facility selects a small number of items having the top combined scores that is other than three. In the example discussed above, the facility would select for prominent display the items having item identifiers "1883823064", "0814403484", and "9676530409". Because the facility in step 907 selects items without regard for their presence in the query result, the facility may select items that are not in the query result. This aspect of this embodiment is particularly advantageous in situations in which a complete query result is not available when the facility is invoked. Such as the case, for instance, where the query server only provides a portion of the items satisfying the query at a time. This aspect of the invention is further advantageous in that, by selecting items without regard for their presence in the query result, the facility is able to select and display to the user items relating to the query even where the query result is empty, *i.e.*, when no items completely satisfy the query. After step 907, these steps conclude.

Figures 10-13 are display diagrams showing examples of considerations used by embodiments of the facility to determine the level of effort expended by the user to select an item from a query result. Figure 10 is a display diagram showing an initial query result display. This display 1010 in browser window 1000 shows the top portion of the first page of a multiple-page query result. The position of scrollbar elevator 1001 at the top of its scrollbar indicates that this web page is scrolled to a position at the top of the web page. This scrolling position is typically the one at which web pages initially display. The display contains the first three items in the query result, items 1011, 1012, and 1013. In general, the first item 1011 is regarded as the easiest item for the user to select, as it is spatially the

nearest item to the beginning of the first page of the query result. Accordingly, in many embodiments, a relatively small increment value is added to rating scores for the selection of this item in this query result. The other items on this display may either be regarded as requiring the same amount of effort to select, since selecting them does not require scrolling the display, or as requiring a slightly higher level of effort to select, because the user must read through one or more other items in the query result to reach these items.

Figure 11 is a display diagram showing a second display of the sample query result shown in Figure 10. This display 1110 shown in browser window 1100 is generated by scrolling down one screen from display 1010 shown in Figure 10, such as by pressing a PageDown key on the keyboard or by clicking the portion of the scrollbar beneath scrollbar elevator 1001. This display 1110 contains the next three items in the query result, items 1111, 1112, and 1113. These three items are typically regarded as requiring more effort for the user to select, as such selection involves scrolling and additional reading not required to select items 1011-1013 shown in Figure 10.

Figure 12 is a display diagram showing a third display of the query result shown in Figure 10. From the relatively low position of scrollbar elevator 1201, it can be seen that this display 1210 shown in browser window 1200 is a screen near the bottom of first page of the query result, which is displayed by further scrolling the page down. Because this larger amount of reading and/or scrolling is required to select one of the displayed items 1211-1213, selecting one of these items typically produces significantly larger increment value than selecting items 1011-1013 or 1111-1113. The display further indicates that this first page of the query result contains the first 25 items of this query result (1221) out of a total of 54 items contained by the entire query result (1222). To reach the next page of this query result, the user can click button 1231 to display a second page of the query result beginning with the 26th item of the query result.

Figure 13 is a display diagram showing a fourth display of the sample query result. Display 1310 is the top of the second page of the sample query result, displayed by clicking button 1231 shown in Figure 12. The display contains items 1311-1313. Selection of any of these items is typically associated with an interval value greater than that for any of the above-discussed items, as reaching this display requires a significant amount of scrolling as well as selecting an intermediate button, also called a "link."

Different embodiments of the facility take various approaches to determining the amount of effort required to select an item in a query result. In some embodiments, the facility determines the amount of effort required based upon the number of items that precede the selected item in the query result. In other embodiments, the facility makes this determination based upon how far down on a query result page the selected item occurs (such as in distance, words, or characters), and/or based upon whether the selected item occurs on a page after the first page of the query result. In other embodiments, the facility uses other approaches to determining the amount of effort needed to select an item from a query result that correspond to various other techniques for navigating a query result.

The facility also uses various mechanisms for performing this determination, including determining the number of items that precede the selected item in the query result, and/or monitoring user interactions that navigate to the selected item within the query result. Such monitoring may encompass monitoring user interface interactions, such as keystrokes, mouse clicks, scroll wheel rotations, and voice commands, as well as by monitoring web server requests corresponding to navigational functions.

In some embodiments, the facility uses stemming techniques to combine rating scores for query terms having the same root as query terms occurring in the query. In different embodiments of the facility, the stemming techniques are incorporated in different ways. As a first example, in the item rating table shown in Figures 3 and 4, the term column containing terms occurring in queries may be replaced with a term root column containing the roots of the terms occurring in queries. For instance, when items are selected from a query containing the terms "human" and "dynamics," the facility would increment scores for item rating table rows containing the term roots "human" and "dynamic" -- the roots obtained by stemming those terms.

In a second example, the facility expands the terms occurring in a query from whose query result an item is selected to all of the different terms that share the same root as the term occurring in the query. For example, for a query containing the term "dynamics," the facility would increment the score for rows with the item rating table containing the terms "dynamic," "dynamics," "dynamism," "dynamist," "dynamistic," and "dynamically."

In a third example, the facility continues to update the item rating table without any use of stemming as described above, but in reading the item rating table, such as in step 804, the facility combines, for each term occurring in the query at issue, the scores for all of

the terms bearing the same root with the terms occurring in the query. For example, if the facility received a query containing the term "dynamics," the facility would combine with the score for this term the scores for the additional terms "dynamic," "dynamism," "dynamist," "dynamistic," and "dynamically," which all share the root "dynam-." Other approaches to
5 utilizing stemming are part of additional embodiments of the facility.

While the present invention has been shown and described with reference to preferred embodiments, it will be understood by those skilled in the art that various changes or modifications in form and detail may be made without departing from the scope of the invention. For example, the facility may be used to rank query results of all types. The
10 facility may use various formulae to determine in the case of each item selection, the amount by which to augment rating scores with respect to the selection. Further, the facility may employ various formulae to combine rating scores into a ranking value for an item. The facility may also use a variety of different kinds of selection actions to augment the rating table, and may augment the rating table for more than one kind of selection action at a time.
15 Additionally, the facility, may augment the rating table to reflect selections by users other than human users, such as software agents or other types of artificial users.